



Internal assessment Report on

Clinic Management System

by

16010121805: **Hardik Panchal**
16010121810: **Soham Patil**
16010121822: **Siddhant Parekh**

Department of Computer Engineering
K. J. Somaiya College of Engineering
(Constituent College of Somaiya Vidyavihar University)
Academic Year 2021 -2022

Problem statement: Clinic Patient management system is introduced to optimize clinics operation. Because of huge changes in management nowadays, management for clinic is important due to the widely spread of technology. This system is proposed for those clinics which are still using paper-based to record down patient record and booking the patient's appointment. The area consists of the user in clinic which is doctor and clinic assistant. Basically, there are no such systems in the clinic. The traditional paper-based management method has caused a lot of problems to the user.

This system will help the user in the clinic in managing the work and booking the appointments efficiently. Also, the Doctor will be able to consult the patient by looking at the log history, and once the consultation is done, doctor can delete that appointment from the log

Chosen Data structure and justification of why the chosen one is the most appropriate to solve the problem:

Data Structure used for this application is Singly Linked List.

It is used to order the patients according to First Come First Serve (FCFS) type, also it is used to sort the type of consultations, i.e., Normal and Emergency type.

Linked List is a flexible Data Structure unlike Stack and Queue, like If an Emergency occurs, top priority will be given to the Emergency Case.

Priority Queue was also an option that we were considering for this application, but each patient in Linked List is treated as a separate node. So, addition and deletion of nodes is more efficient in Linked List as compared to Priority Queue.

As mentioned earlier, if an Emergency Case occurs then we can add a new node in between for the types of consultation including normal consultation.

Conceptual solution and allied details: The flowchart for the application is mentioned below It says, user can either enter doctor details or patient details.

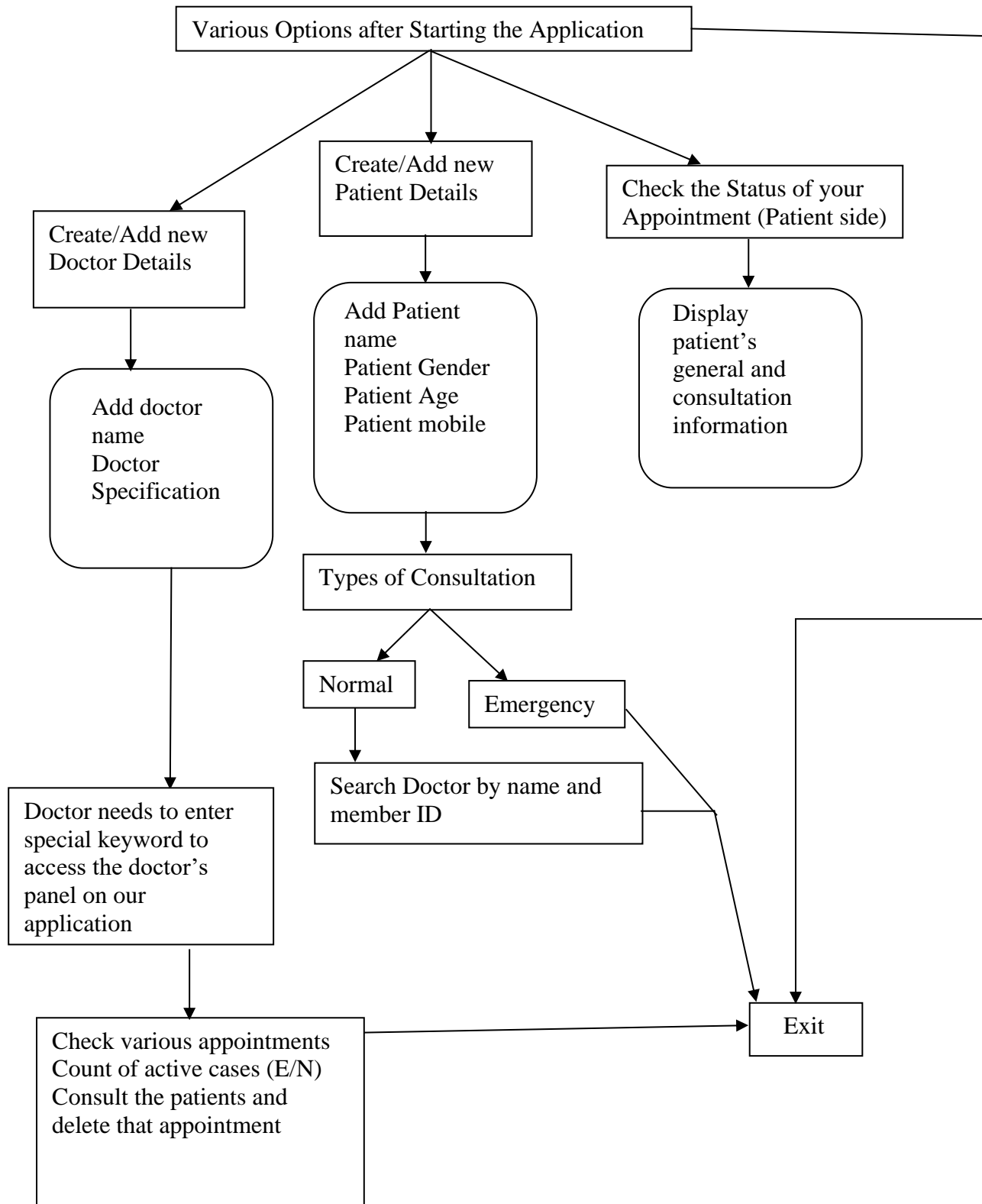
After entering the details of doctor and patient, the system will automatically generate member-Id for both using **Hash Function**. After entering the details, patient can check their details in status section by entering their member-Id.

Patient can choose the type of consultation i.e., Normal and Emergency, after selecting the Consultation, search for the doctor by name or member-Id.

Doctor has a special keyword wherein he can access the system for checking various appointments, deleting various appointments after consulting them and checking the number of active cases for consultation.

Once all the operations are over, patient and doctor can exit from the application. Once they again enter the details, System will again generate new Member-Id for them using Hash Function.

Clinic Management System



Source code and output screenshots:

Source Code:

Link to online file –

https://drive.google.com/drive/folders/1sObKscCu5FNVVpVUrMt_HbaqbzwbJjxZ?usp=sharing

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
int total_cases = 0, Normal = 0, Emergency = 0;
int opt;
struct pa_node // Patient Node
{
    char name[100], mode_of_admission[100], gender[100];
    char mobile[10];
    char med[200], drugs[100], other[100], operations[100], dia[100],
current[100], doctor_visit[100], modeofconsultation[100], age[3];
    int height, weight, token;
    struct pa_node *pat1;
    struct doc_node *doctor;
} * t, *tem, *head = NULL;
struct doc_node // Doctor N0de
{
    char name[100], dept[100];
    struct doc_node *pat, *id;
    struct pa_node *pat1;
} * p, *temp;
struct doc_node *a[26];
void make_it_NULL() // This is a function which makes every element of a pointer
is equal to NULL
{
    int i;
    for (i = 0; i < 26; i++)
        a[i] = NULL;
}
void new_doctor_entry(char a1[]) // This functions is a entry portal to the new
doctors to enter into the data base of the system
{
    int m;
    struct doc_node *newnode, *p;
    newnode = (struct doc_node *) (malloc(sizeof(struct doc_node)));
    newnode->pat = NULL;
    newnode->pat1 = NULL;
    printf("Department of work: ");
```

```

scanf("%s", newnode->dept);
a1[0] = tolower(a1[0]);
strcpy(newnode->name, a1);
m = a1[0] - 97; // hash function
if (a[m] == NULL)
{
    a[m] = newnode;
}
else
{
    p = a[m];
    while (p->pat != NULL)
        p = p->pat;
    p->pat = newnode;
}
printf("\n\nDoctor %s this is your member ID: %x\n\n", newnode, newnode-
>name);
}

struct doc_node *doctor_find() // This function prints the all possible doctors
it can be with input as name something like search engine
{
    int m, asd;
    char a1[100];
    int flag = 0, flag1;
    struct doc_node *aacc;
    printf("\n\nSearch doctor \n1. By Name\n2. By member ID\n\nEnter your choice
: ");
    scanf("%d", &flag1);
    if (flag1 == 1)
    {

        printf("Enter the doctor's name : ");
        scanf("%s", a1);
        m = a1[0] - 97;
        if (a[m] == NULL)
        {
            printf("Doctor Not found\n1. Do you want to try again\n2.
Exit\n\nEnter your choice : ");

            scanf("%d", &asd);
            if (asd == 1)
                doctor_find();
            else
                return NULL;
        }
    }
}

```

```

else
{
    p = a[m];
    while (p->pat != NULL)
    {
        if (strcmp(p->name, a1) == 0)
        {
            printf("Doctor name : %s\nDoctor Dept. : %s\nDoctor member ID : %x\n", p->name, p->dept);
            flag++;
        }
        p = p->pat;
    }
    if (strcmp(p->name, a1) == 0)
    {
        printf("Doctor name : %s\nDoctor Dept. : %s\nDoctor member ID : %x\n", p->name, p->dept);
        flag++;
    }
    if (flag == 0)
    {
        printf("The doctor doesn't exist!!");
        printf("\n1. Do you want to try again\n2. Exit\n\nEnter your choice : ");
        scanf("%d", &asd);
        if (asd == 1)
            doctor_find();
        else
        {
            if (asd == 2)
                return NULL;
        }
    }
    else
    {
        printf("Enter the doctor member ID : ");
        scanf("%x", &aacc);
        return aacc;
    }
}
}
else
{
    if (flag1 == 2)
    {
        printf("Enter the doctor member ID : ");
    }
}

```

```

        scanf("%x", &aacc);
        return aacc;
    }
}
return aacc;
}

void doc_status(struct doc_node *doctor) // This is a profile report or the
working details of the doctor
{
    struct pa_node *coun_pa, *patient;
    printf("\n\nDoctor name : %s\n", doctor->name);
    printf("Doctor member ID : %x\n", doctor);
    p = doctor;
    coun_pa = p->pat1;
    int m = 0;
    while (coun_pa != NULL && p != NULL && p->pat1 != NULL)
    {
        coun_pa = coun_pa->pat1;
        m++;
    }
    printf("The remaining number of patients to consult are %d", m);
}

void show_remain(struct doc_node *doctor)
{
    struct pa_node *coun_pa, *patient;
    p = doctor;
    coun_pa = p->pat1;
    int m = 0;
    while (coun_pa != NULL && p != NULL && p->pat1 != NULL)
    {
        coun_pa = coun_pa->pat1;
        m++;
    }
    printf("\n\nAfter completion of current consultation the remaining number of
patients to consult are %d", m);
}

void patient_log(struct doc_node *doctor) // This is the log report of the
patients who are remaining for the consultation
{
    struct pa_node *coun_pa, *patient;
    printf("\n\nDoctor name : %s", doctor->name);
    printf("\nDoctor member ID : %x\n\n", doctor);
    p = doctor;

```

```

    coun_pa = p->pat1;
    int m = 0;
    while (coun_pa != NULL && p != NULL && p->pat1 != NULL)
    {
        printf("Patient's name : %s\nPatient's ID : %x\n\n", coun_pa->name,
coun_pa);
        coun_pa = coun_pa->pat1;
    }
}

bool check_mobile(char mobile[])
{
    int i = 0;
    while (i < 10)
    {
        if (mobile[i] == -1)
        {
            printf("\nPlease enter a valid mobile number.\n");
            return true;
        }
        i++;
    }
    return false;
}

void pat_entry_doctor() // This function makes an appointment with the respetive
consultant
{
    char a[100], b[100];
    struct pa_node *patient;
    struct doc_node *l;
    patient = (struct pa_node *) (malloc(sizeof(struct pa_node)));
    t = (struct pa_node *) (malloc(sizeof(struct pa_node)));
    patient->pat1 = NULL;
    printf("\n\n***** Patient's GENERAL INFORMATION *****\n");
    printf("\nEnter patient's name : ");
    scanf("%s", patient->name);
    printf("\nEnter patient's gender : ");
    scanf("%s", patient->gender);
    int i = 0;
    while (i < 10)
    {
        patient->mobile[i] = -1;
        i++;
    }
    do

```



```

{
    printf("\nEnter patient's mobile number : ");
    scanf("%s", patient->mobile);
} while (check_mobile(patient->mobile));
printf("\n***** PATIENT INFORMATION *****\n");
printf("\nPatient's Age : ");
scanf("%s", patient->age);
printf("\nWhat type of consultation?\n\n1.Normal\n2.Emergency\n\nEnter your
choice: ");
int as;
scanf("%d", &as);
if (as == 1)
{
    patient->doctor = doctor_find();
    l = patient->doctor;

    if (l != NULL)
    {
        strcpy(patient->doctor_visit, patient->doctor->name);
        t = l->pat1;
        if (t == NULL)
        {
            t = patient;
            l->pat1 = t;
            printf("The patient member ID is : %x\n", patient);
            Normal++;
        }
        else
        {
            while (t->pat1 != NULL)
                t = t->pat1;
            t->pat1 = patient;
            printf("The patient member ID is : %x\n", patient);
            Normal++;
        }
    }
    else
        printf("Doctor with such ID doesn't exist!!\n");
}
else
{
    Emergency++;
    if (head == NULL)
    {
        head = patient;
    }
}

```

```

        else
        {
            tem = head;
            while (tem->pat1 != NULL)
                tem = tem->pat1;
            tem->pat1 = patient;
        }
    }
}

void del(struct doc_node *doctor) // After completion this function deletes
the patient node visited
{
    struct pa_node *node;
    if (doctor->pat1 == NULL)
        printf("The patient list is empty...");
    else
    {
        node = doctor->pat1;
        doctor->pat1 = node->pat1;
        free(node);
    }
}

int main()
{
    int choice;
    char g[100];
    make_it_NULL();
    do
    {
        total_cases = Normal + Emergency;
        printf("\n\nStatistics of Clinic\n");
        printf("\n\nTotal number active cases : %d\nTotal number of normal
consultations : %d\nTotal number of emergency cases : %d\n", total_cases, Normal,
Emergency);
        printf("\nWhat you want to do?");
        printf("\n\n1. Book an appointment for patient\n2. Create a new doctor
entry\n3. Status of patient's appointment \n4. Exit\n100. Doctor options\n\nEnter
your choice : ");
        scanf("%d", &choice);
        switch (choice)
        {
            case 1:
                printf("\nWelcome to patient admission portal: ");
                pat_entry_doctor();

```

```

        break;
    case 2:
        printf("\n\nEnter doctor's name: ");
        scanf("%s", g);
        new_doctor_entry(g);
        break;
    case 3:
        printf("\n\nEnter patient's member ID: ");
        struct pa_node *coun_pa, *patient;
        scanf("%x", &patient);
        printf("\n Name : %s\n Gender : %s\n Mobile number : %s\n Age : %s \n
Consultant Doctor : %s\n Consultant Doctor's member ID : %x\n", patient->name,
patient->gender, patient->mobile,
        patient->age, patient->doctor_visit, patient->doctor);
        p = patient->doctor;
        coun_pa = p->pat1;
        int m;
        m = 1;
        while (coun_pa != patient && p != NULL && p->pat1 != NULL)
        {
            coun_pa = coun_pa->pat1;

            m++;
        }
        printf(" Token number %d\n", m);
        break;
    case 4:
        printf("Have a great day!!! ... \n");
        break;
    case 100: // By using this we enter the doctor mode from patient mode
        printf("\n\nDoctor what would you like to do?");
        int choo = 0;
        do
        {
            printf("\n\n1. Consultation completion\n2. Check remaining
patients\n3. Patient log for remaining patients\n4. Exit\n\nEnter your choice :
");

            scanf("%d", &choo);
            if (choo != 4)
            {
                printf("\n\nPlease enter your member ID : ");
                scanf("%x", &temp);
            }
            switch (choo)
            {
                case 1:

```

```
        del(temp);
        Normal--;
        show_remain(temp);
        break;
    case 2:
        doc_status(temp);
        break;
    case 3:
        patient_log(temp);
        break;
    case 4:
        printf("Have a great day!!! ... \n");
        break;
    }
} while (choo != 4);
} while (choice != 4);

return 0;
}
```

Output:

Creating a new doctor entry –

```
Statistics of Clinic
```

```
Total number active cases : 0
```

```
Total number of normal consultations : 0
```

```
Total number of emergency cases : 0
```

```
What you want to do?
```

```
1. Book an appointment for patient
```

```
2. Create a new doctor entry
```

```
3. Status of patient's appointment
```

```
4. Exit
```

```
100. Doctor options
```

```
Enter your choice : 2
```

```
Enter doctor's name: abc
```

```
Department of work: surgeon
```

```
Doctor abc this is your member ID: e21678
```

Creating a new patient entry –

What you want to do?

1. Book an appointment for patient
2. Create a new doctor entry
3. Status of patient's appointment
4. Exit
100. Doctor options

Enter your choice : 1

Welcome to patient admission portal:

***** Patient's GENERAL INFORMATION *****

Enter patient's name : pqr

Enter patient's gender : male

Enter patient's mobile number : 742136980

***** PATIENT INFORMATION *****

Patient's Age : 21

What type of consultation?

- 1.Normal
- 2.Emergency

Enter your choice: 1

Search doctor

1. By Name
2. By member ID

Enter your choice : 2

Enter the doctor member ID : e21678

The patient member ID is : e228c8

Creating another new patient entry –

Statistics of Clinic

Total number active cases : 1
Total number of normal consultations : 1
Total number of emergency cases : 0

What you want to do?

1. Book an appointment for patient
2. Create a new doctor entry
3. Status of patient's appointment
4. Exit
100. Doctor options

Enter your choice : 1

Welcome to patient admission portal:

***** Patient's GENERAL INFORMATION *****

Enter patient's name : mno

Enter patient's gender : female

Enter patient's mobile number : 2136549790

***** PATIENT INFORMATION *****

Patient's Age : 45

What type of consultation?

- 1.Normal
- 2.Emergency

Enter your choice: 1

Search doctor

1. By Name
2. By member ID

Enter your choice : 2

Enter the doctor member ID : e21678

The patient member ID is : e24450

Checking appointment status of patients -

Statistics of Clinic

Total number active cases : 2

Total number of normal consultations : 2

Total number of emergency cases : 0

What you want to do?

1. Book an appointment for patient
2. Create a new doctor entry
3. Status of patient's appointment
4. Exit
100. Doctor options

Enter your choice : 3

Enter patient's member ID: e228c8

Name : pqr

Gender : male

Mobile number : 742136980

Age : 21

Consultant Doctor : abc

Consultant Doctor's member ID : e21678

Token number 1

Statistics of Clinic

Total number active cases : 2

Total number of normal consultations : 2

Total number of emergency cases : 0

What you want to do?

1. Book an appointment for patient
2. Create a new doctor entry
3. Status of patient's appointment
4. Exit
100. Doctor options

Enter your choice : 3

Enter patient's member ID: e24450

Name : mno

Gender : female

Mobile number : 2136549790

Age : 45

Consultant Doctor : abc

Consultant Doctor's member ID : e21678

Token number 2

Options for doctor –

Statistics of Clinic

Total number active cases : 2

Total number of normal consultations : 2

Total number of emergency cases : 0

What you want to do?

1. Book an appointment for patient
2. Create a new doctor entry
3. Status of patient's appointment
4. Exit
100. Doctor options

Enter your choice : 100

Doctor what would you like to do?

1. Consultation completion
2. Check remaining patients
3. Patient log for remaining patients
4. Exit

Enter your choice : |

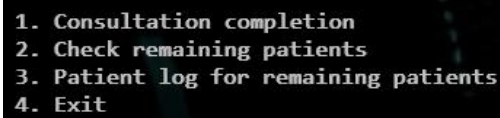
Checking number remaining patients for doctor –

```
Doctor what would you like to do?  
  
1. Consultation completion  
2. Check remaining patients  
3. Patient log for remaining patients  
4. Exit  
  
Enter your choice : 2  
  
Please enter your member ID : e21678  
  
Doctor name : abc  
Doctor member ID : e21678  
The remaining number of patients to consult are 2
```

Checking log of remaining patients for doctor –

```
1. Consultation completion  
2. Check remaining patients  
3. Patient log for remaining patients  
4. Exit  
  
Enter your choice : 3  
  
Please enter your member ID : e21678  
  
Doctor name : abc  
Doctor member ID : e21678  
  
Patient's name : pqr  
Patient's ID : e228c8  
  
Patient's name : mno  
Patient's ID : e24450
```

Consultation completion for doctor –

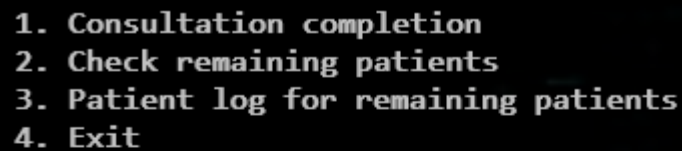
- 
1. Consultation completion
 2. Check remaining patients
 3. Patient log for remaining patients
 4. Exit

Enter your choice : 1

Please enter your member ID : e21678

After completion of current consultation the remaining number of patients to consult are 1

Checking remaining log of patients –

- 
1. Consultation completion
 2. Check remaining patients
 3. Patient log for remaining patients
 4. Exit

Enter your choice : 3

Please enter your member ID : e21678

Doctor name : abc

Doctor member ID : e21678

Patient's name : mno

Patient's ID : e24450

Test case for invalid input –

Invalid phone number –

What you want to do?

1. Book an appointment for patient
2. Create a new doctor entry
3. Status of patient's appointment
4. Exit

Enter your choice : 1

Welcome to patient admission portal:

***** Patient's GENERAL INFORMATION *****

Enter patient's name : xyz

Enter patient's gender : male

Enter patient's mobile number : 745896

Please enter a valid mobile number.

Enter patient's mobile number : asdf

Please enter a valid mobile number.

Enter patient's mobile number : 7412589630

***** PATIENT INFORMATION *****

Patient's Age : |

Invalid doctor name and member ID –

```
What type of consultation?
1.Normal
2.Emergency

Enter your choice: 1

Search doctor
1. By Name
2. By member ID

Enter your choice : 1
Enter the doctor's name : asd
The doctor doesn't exist!!
1. Do you want to try again
2. Exit

Enter your choice : 1

Search doctor
1. By Name
2. By member ID

Enter your choice : 2
Enter the doctor member ID : 123
Doctor with such ID doesn't exist!!
```

Conclusion and take away from this exercise: Hence, we analyzed and implemented the Problem definition on Clinic Management System by using appropriate data structure i.e., Linked List.

As Linked List is a flexible data structure, we could efficiently use it in our implementation. Also, we understood the use of Hash Function to generate Unique Member-Id for Doctors and Patients, which will uniquely generate new Member-Id for each user whenever user enters his/her data.

As the traditional paper-based management method has caused a lot of problems to the user as well as to the doctor, so this clinic management system will help the users to some extent and will ease their work.